T.C.ブローウェル

任意の順列パズルを解く

学士論文 2016年6月18日 監修者

R.M.ファン・ルイク博士

ライデン大学数学研究所

内容

1　はじめに

この学士論文は、ルービックキューブから着想を得ている。ルービックキューブは有名な順列パズルであり、その解法に関する数学的研究がいくつか行われている（例えば、[4]を参照）。

最初の設定は、4次元のルービックキューブを考え、それに対する一般的な解を書くというものだった。しかしその過程で、我々はより一般的な問題設定に移行した。次節でより正確に述べるが、本稿では、任意の順列パズルに対する一般解を書くためのアルゴリズムを提案する。我々の結果は第12節にまとめられている。

2　数学的な の定式化

*表記*本稿では、Sym(n)は

n要素。

定義2.1.TをSym(n)の部分集合とする。Tの**逆閉包を** $\hat{T} = T \cup \{t^{-1} : t \in T\}$ と定義する。$T = \hat{T}$のとき、Sym(n)の部分集合Tは*逆閉*と呼ばれる。

定義2.2.TをSym(n)の部分集合とする。Tの**語群は**$\hat{T}$上の自由群である

。

Tの単語群に含まれる任意の要素tは、$t = (t_1, q_1, t_2, q_2, \ldots, t_k, q_k)$ の並びで表すことができ、t_i は$\hat{T}$に、q_i は $\{1, 1\}$ に含まれる。sをSym(n)とする。Tの単語群の要素をTの**単語と**呼ぶ。単語 $t = (t_1, q_1, t_2, q_2, \ldots, t_k, q_k)$ を持つTの単語群における $\prod_{i=1}^{k} t_i^{q_i} = s$ とする。は$t_1^{q_1} t_2^{q_2} \cdots t_k^{q_k}$と書き、$t$は$s$を書く。$t_i^{q_i}$ は文字と呼ばれる。もし $t = (t_1, q_1, \ldots, t_m, q_m)$ および $u = (u_1, p_1, \ldots, u_n, p_n)$ はT内の2つの単語である。

とは

$$(t_1, q_1, \ldots, t_m, q_m, u_1, p_1, \ldots, u_n, p_n)$$

順列パズルでは，Sym(n) の部分集合 T と $s \in \langle T \rangle$ *が与えられる*．並べ替えパズルの目的は、T に含まれる s ができるだけ短い単語を見つけることである。本稿では以下のことを行う：

- 第3節から第5節では、Sym(n)の任意の部分集合Tに対して、任意のs
  $\in \langle \rangle$
  TをTの単語として書くための最小の長さがどの程度かを調べる。

- 第6節から第11節では、Sym(n)の任意の部分集合Tに対して、任
  $\in \langle \rangle$
  意のs TをTの単語として書くことができるアルゴリズムを構築する。

3 nにおける下限値

このセクションでは、任意の s $\in \langle \rangle$ TをT
Sym(n)の任意の部分集合Tについて、nの多項式長で計算することはできない。

定理3.1. *各nについて、T_n がSym(n)の部分集合である数列($T_1, T_2, \ldots, T_n$, ...)、および、各s_n がT_n にある要素の数列($s_1, s_2, \ldots, s_n, \ldots$)が存在し、$s_n$ がnの多項式で囲まれる長さのT_n の単語として書けないようなものである。*

証明。 $p_1 < p_2 < p_3 < \ldots$ を素数とする。i_n を最大の自然数

$\sum_{j=1}^{i_n} p_j \leq n$ となるような数。

Sym(n)のq_n を、長さ$p_1, p_2, \ldots p_i$ n.

Tとする $= \{q_n\}$.sを選ぶ。$_n = (q_n)^{|\langle T_n\rangle|/2}$.

。$_n$

s_n をT_n の単語として書くには、T_n から少なくとも$|\langle T_n\rangle|/2$ の文字が必要である。

q_n の $/ \langle T_n \rangle |$ は$\prod_{j=1}^{i_n} p_j$ に等しい。すべてのiについて、$p_i \geq 2$である。

q_n の次数が 2^i n より大きいこと。

ここで、p_i nが少なくともp≦となる最大の素数pであることを証明する。

√ $n.p_i$ nがp≦nより小さいとする。 $\bar{n}$.そのとき、i_n は最大ではない。

$p_{i_n+1} \leq \sqrt{n}$、したがって $i_n + 1 \leq \sqrt{n}$、したがって $\sum_{j=1}^{i_n+1} p_j \leq \sum_{j=1}^{[\sqrt{n}]} \sqrt{n} \leq n$。

$p_i\ n \geq p$ したがって、i_n は以下の数より大きいか等しいと結論づけられる。

以下の素数の $\bar{n}$.

素数の定理により、nが十分に大きい場合、$i_n \geq$であることがわかる。

$\frac{1}{2}\frac{\sqrt{n}}{\log(\sqrt{n})}$ 1]の9ページを参照。したがって、$/ \langle T_n \rangle| \geq 2^{\frac{1}{2}\frac{\sqrt{n}}{\log(\sqrt{n})}}$, そして、我々は少なくとも以下を必要とする。

$2^{2^{\frac{1}{2}\frac{\sqrt{n}}{\log(\sqrt{n})}}}$ letters from T_n to write s_n, which is not bounded by polynomial in n. □

4 グループ内の上限 順

この節では群次数の観点から上限を証明する。ここでは[2]の3.1節にあるような群のケイリーグラフの概念を用いる。

定義4.1. Sym(n)の部分集合Tの**Cayley-graph** $W_T = (T, E)$は、頂点の集合がTの要素の集合に等しい無向グラフである。二つの頂点 g_1 と g_2 は、g_1 t = g_2 となるような要素 t が$\hat{T}$ に存在する場合にのみ接続される。

次のレンマはケイリーグラフの理論から導かれたもので、群順の観点から単語の長さの上限を決めるのに役立つ。

定理4.1. Sym(n)*の反比例的に閉じた部分集合Tが与えられたとき⟨T⟩は、T中の単語として、最大でも* $\#\langle T\rangle$ *の長さで書くことができる。*

*証明。*一般性を損なうことなく、$1 \notin T$と仮定してもよい。Tが空であれば、この文が真であることを示すには、空の単語で十分である。

もし$\#T$が1なら、Tは逆閉なので、T*の要素は*次数2でなければならない。今、Tの唯一の自明でない要素は1文字の単語として書くことができ、この文は真である。

ここで#Tと仮定する。≥2とすると、ケイリーグラフの次数は少なくとも2である。

dはケイリーグラフの次数を表す。ケイリーグラフの理論から

ケイリー・グラフの頂点連結性は少なくとも$\frac{2(d+1)}{}$、定理3.7を参照。

の[2]であるから、この場合は少なくとも2である。W^i をW_T の頂点の集合とする。

^{i}同様に、W^i 、 $\langle T$ ↪Pe_27E9 の要素のうち、長さが最大でも i の T

の単語として書くことができるものを含む。 T T

for $i < \frac{\#\langle T \rangle}{2}$.

$W^0 = 1$である。ここで、$i + 1 < \frac{\langle T \rangle}{}$ の自然数iについて、次のように仮定する。 T 2

$\#W_T^i \geq 2i + 1$.もし $_T = \langle T \rangle$を証明するものは何もない。 $= \#\langle T \rangle - 1$

W^i

もし#W^i T

の場合、欠けている頂点は $W_T^{i+1} = \langle T \rangle$ にあり、ステートメントが成立する。そこで

はW^i の外側にある少なくとも2つの頂点である。W^i は少なくとも1つの T T

の頂点の連結性（例えばv_1 、このv_1 ）はW^{i+1} 。$_T$

は少なくとも2であり、またv_1 を取り除いたW_T は連結しているので、別の頂点が存在する。

v_2 W^i の外側は W^{i+1} であり、$_T$ したがって $\#W_T^{i+1} \geq \#W_T^i + 2 \geq 2(i+1) + 1$ である。

私たちは今、2つのケースを見極めようとしている。

- $j = \frac{\langle T \rangle}{} - 1$のとき、$\#W^j \geq \#\langle T \rangle - 1$が成り立つ。したがって W^j にない要素は、Tの中にせいぜい1つしかない。しかし、もしこの要素が存在すれば、Cayley-graphは連結しているので、もう一文字使って書くことができる。$\frac{\langle T \rangle}{2}$したがって、 $\langle T \rangle$ のどの要素も、T の単語として書くことができる。

- ここでkを整数$\frac{\langle T \rangle - 1}{2}$ 。次のことが成り立つ。 $\#W_T^k = \langle T \rangle$。というわけで、ステートメントが成立する。

これで証明は終わった。 □

このセクションの最後に、これが群順位の観点から可能な限り低い上限であることを示す。

定理4.2. *T を Sym(n) の部分集合とする。Tのどの要素も、長さが最大でも $\frac{|\langle T\rangle|}{2}$ 、Tの単語として書くことができる。また、すべての n に対して、$|\langle T_n\rangle| \geq n$ となる Sym(n) の部分集合 T_n が存在し、$\langle T_n\rangle$ の要素 s は、s が次のようになることはできない。$\frac{|\langle T_n\rangle|}{2} - 1$ 以下の長さのT_n の単語として書かれる。*

証明。 最初の記述は、まさにLemma 4.1である。2番目の説明を証明するために、$s = (1 \cdots n) \in \mathrm{Sym}(n)$ とし、$T_n = \{s\}$ とする。このとき、$|\langle T_n\rangle| = n$ があり、要素 $s^{\lfloor n/2 \rfloor}$ を書くには少なくとも $\frac{n}{2} - 1$ 文字が必要である。 □

5 Sym(n)の例

このセクションでは、Sym(n)の要素を$T = \{(12), (1 \cdots n)\}$.

次の定義が必要だ。

定義5.1. 順列$s \in \mathrm{Sym}(n)$が与えられたとき、各$m \in \{1, \ldots, n\}$ に対して、mの元の位置までの距離を次のように定義する。

$$\min(|s(m) - m|, n - |s(m) - m|).$$

$d(s)$で示されるsの総処置は、元の位置までの距離の合計、すなわち

$$d(s) = \sum_{m=1}^{n} \min(|s(m) - m|, n - |s(m) - m|).$$

tを順列$(1 - - - n)$とする。sの**巡回**処分は$\min_{j\in \mathbf{Z}} (d(t^j s))$である。

次の証明では、次の事実を用いる。x、t、nを$0 \leqq x < n$、$0 \leqq t < n$の自然数とすると、次が成り立つ。

$\min(|x - t|, n - |x - t|) = \min(x - t \bmod n, n - (x - t) \bmod n)$ ここで a

$\bmod n$ は $0, \ldots, n$ $\{ \quad - \}1$ の残差クラスの代

表である。

a in $\mathbf{Z}/n\mathbf{Z}$.

定理5.1. *順列$s \in \mathrm{Sym}(n)$が与えられたとき、sの巡回配置は最大でも$\frac{n^2}{4}$。また、nが奇数であるすべてのnについて、sの巡回配置が少なくとも$\frac{n^2-1}{4}$であるような$s \in \mathrm{Sym}(n)$が存在する。*

証明。次のように書く。

$$\begin{aligned}
\sum_{j=1}^{n} d(t^j s) &= \sum_{j=1}^{n}\sum_{m=1}^{n} \min(|t^j s(m) - m|, n - |t^j s(m) - m|) \\
&= \sum_{m=1}^{n}\sum_{j=1}^{n} \min(|t^j s(m) - m|, n - |t^j s(m) - m|) \\
&= \sum_{m=1}^{n}\sum_{i=1}^{n} \min(i, n - i)
\end{aligned}$$

この最後の和は、nが奇数なら$\frac{n^3-n}{4}$、nが偶数なら$\frac{n^3}{4}$である。次のようになる。

a j は、$d(t^j s)$ が最大でも $\frac{n^2}{4}$ となる。

n を奇数とし、$f(m) = 2m - 1 \bmod n$ で定義される関数 $f : \mathbf{Z/nZ} \longrightarrow \mathbf{Z/nZ}$ を考える。n は奇数なので、2の乗算は両射影であり、写像 $x \mapsto x - 1$ との合成も両射影である。

fは両対称なので、Sym(n)の順列 s_f を定義する。この順列は

tion s_f はmを$f(m)$に写像する。ここで、s_f の総処分を計算する。

$$
\begin{aligned}
d(s_f) &= \sum_{m=1}^{n} \min(|s_f(m) - m|, n - |s_f(m) - m|) \\
&= \sum_{m=1}^{n} \min(|f(m) - m|, n - |f(m) - m|) \\
&= \sum_{m=1}^{n} \min(|((2m - 1) \bmod n) - m|, n - |((2m - 1) \bmod n) - m|) \\
&= \sum_{m=1}^{n} \min(m - 1, n - (m - 1)) \\
&= \frac{n^2 - 1}{4}
\end{aligned}
$$

ここで、s_f の環状配置がs_f の全配置と等しいことを示す。jを**Z**とし、tを順列$(1 - - n)$とする。^{j}順列$t\ s^j{}_f$ は、mを$f_j\ (m) = f(m) + j \bmod n = 2m - 1 + j \bmod n$に写す写像$f_j$ によって特徴づけられる。

$$
\begin{aligned}
d(t^j s) &= \sum_{m=1}^{n} \min(|s_{f_j}(m) - m|, n - |s_{f_j}(m) - m|) \\
&= \sum_{m=1}^{n} \min((|(2m - 1 + j) \bmod n) - m|, n - |((2m - 1 + j) \bmod n) - m|) \\
&= \sum_{m=1}^{n} \min(|m - 1 + j| \bmod n, n - (|m - 1 + j| \bmod n)) \\
&= \sum_{m=1}^{n} \min(m, n - m) \\
&= \frac{n^2 - 1}{4}
\end{aligned}
$$

したがって、nが奇数であるすべてのnについて、sの環状配置が少なくとも$\frac{n^2-1}{4}$ であるような$s \in \mathrm{Sym}(n)$が存在することがわかる。 □

定義5.2. $\mathrm{Sym}(n)$の部分集合Tに対して、最小単語長関数γ_T は、T中のsに対して、$\gamma_T(s)$がsを書くT中の単語の最小の長さであるようなT上の関数である。

定義5.3. ある領域Dから自然数への2つの関数をfとgとする。ある定数cに対して、すべての$d \in D$に対して$c \cdot f(d) > g(d)$が成り立つとき、関数f

はΩ(g)であるという。

定理5.2.$T_n = \{(12), (1 \ldots n)\} \subset$ を持つ数列 $(T_2, T_3 \ldots, T_n, \ldots)$ が与えられる。
$\mathrm{Sym}(n)$, $s_n \in \mathrm{Sym}(n)$ を持つシーケンス $(s_2, s_3, \ldots, s_n, \ldots)$ が存在する。

関数

$$\Gamma : \{3, 4, . . .\} \to \mathbb{N}$$
$$n \mapsto \gamma_T \mathrm{n}\,(s\,)_n$$

は$\Omega(\mathrm{n}^2)$で

ある。

*証明。*レンマ5.1により、n*が*奇数であるすべてのn*について、次の*ような$s_n \in \mathrm{Sym}(n)$ が存在する。

$_n$*の*環状配置は少なくとも $\frac{n-1^2}{4}$ である。ここで

$(s_3 , ... , s_{2n+1} , ...)$ とする。n *を* **N** とする。$(t_1 , ... , t_k)$ を T_n の s_n に対する単語とする。

$d_i = {}^{Qi}$ と定義する。 t_i と定義する。数列 $(d_1 , d_2 , ... , d_k)$ を考える。a_i を環状 $_{j=1}$

d_i 。$_k$ $= s_n$ *となる。*$_k$ $\geq \frac{n2-1}{4}$.もしt_{i+1} $= (1 - - - n)$

またはその逆であれば、 $a_{i+1} = a_i$. $t_{i+1} = (12)$であれば、2つの要素だけが移動され、元の位置との距離は最大でも1つしか増えないので、 $a_{i+1} \leq a_i + 2.t_i$ と $\frac{t_{i+1}}{4}$ の両方が(12)に等しい場合， $a_{i+2} = a_i$. したがって， $a_{i+2} \leq a_i + 2.a_0 = 0$および$a_1 \leq 2$であるから、$a_k$ は最大でもk + 1であり、したがってk*は*少なくとも$^{n\,2-1}$ - 1である。

$n \geqq 4$でn*が*偶数の場合、$s_{n-1} \in \mathrm{Sym}(n)$ *となり*、n - 1は奇数である。

を表す単語は、少なくとも - 1 = - 3 の長さを持つ。 は少なくとも$^{(n-1)}$ - 1 = $^{n-2n}$ - ^{3} の長さを持つ。したがって $_{\text{n-1}}$ $\frac{2}{4}$ $\frac{2}{4}$ $\frac{}{4}$

Γは$\Omega(\mathrm{n}^2)$

である。 □

定理5.3.$n>1$を**N**と*し、T を*$\{(12), (1 . . n)\} \subset \mathrm{Sym}(n)$*の集合とする。ここで、すべての*$s \in \mathrm{Sym}(n)$*は、*$\frac{3n^2}{2}$ *以下の文字を使って*T*の単語として書くことができる。*

この定理の証明を行う前に、$s \in \mathrm{Sym}(n)$を、指定された文字数を用いてT*の*単語として書き出すアルゴリズムの概略を示す。このアルゴリズムはs*から*始まり、2,3, ...,n*を1に対し*てソートする。.,nを1に対してソートする。各段階において、アルゴリズムは並べ替えに使われた順列を出力し、並べ替えが終わると、出力された要素の逆数を逆順に並べた

ものが、*Tにおけるsの*単語となる。

アルゴリズムに不可欠なのは、1に対して相対的にソートすることである。つまり、2から始まるアルゴリズムは、2の次に1を動かす文字を出力する。続けて、アルゴリズムは2の隣に3を移動させる文字を出力する。nが1の隣に移動すると、すべての数字は1に対してソートされる。1の位置に応じて、文字$(1\ .\ \ldots\ n)$のみからなるシーケンスがすべての数字を元の位置に移動させ、アルゴリズムは終了する。

アルゴリズムの定義により、*iより*小さいすべての数はすでに1に対してソートされていると仮定する。次にアルゴリズムは、iが1に対してソートされるまで、文字(12)と$(1\ .\ \ldots\ n)$を交互に出力する。文字(12)が適用されたとき、*iより*小さい文字はすべて1と2の元の位置になかったので、*iより*小さいすべての数はまだ1に対してソートされている。

アルゴリズムの複雑さは、以下の定理5.3の証明で与えられる。

*証明。*このアルゴリズムは2つの部分で構成されている。

1. すべての数 2, ..., n を 1 に対して相対的に並べ替える。

2. 1、ひいてはすべての数字を元の位置に戻す。この2つ

のパートを以下に分析する。

1. 数iを$1 \leqq i < n$でソートする場合、アルゴリズムは次のようになる。
   $(1 \ldots n)$または$(n \ldots 1)$のいずれかを最大$\frac{n}{2}$回出力することで、数iは次のようになる。
   このアルゴリズムは、(12)と$(1 \ldots n)$の文字の組み合わせを$n - i$回まで出力する。iをソートする場合、アルゴリズムは最大で$\frac{n}{2} + 2(n - i)$の文字を出力する。1に対してnまでのすべての数をソートする場合、アルゴリズムは最大で次のように出力する。

$$\sum_{i=2}^{n}\left(\frac{n}{2} + 2n - 2i\right) \leq \frac{5n^2}{2} - n - n = \frac{3n^2}{2} - n$$

の手紙だ。

2. 1を返し、すべての数を元の位置に戻す場合、このアルゴリズムは最大で$\frac{n}{2}$、（$1 \ldots n$）または（$n \ldots 1$）のいずれかを出力する。

$\square$

この2つの部分を組み合わせると、このアルゴリズムは$\frac{3n^2}{2}$文字以下を出力することがわかる。

*備考*5.1.上述のアルゴリズムは最適ではない。

6 シュライアの基本コンセプト シムズ

次の2つの節は、[3]の第4章第1節と第2節に記述されているSchreier-Simsアルゴリズムに基づいている。ここでの証明の一部は、Seressの素晴らしい本から得たものであり、わずかな修正を加えただけである。例えば、我々は基底の要素が$\{1, \ldots, n\}$ *の任意の部分集合である*。Schreier-Simsアルゴリ

ズムのさらなる最適化については、本書の第4章3節以降に記述されている。

このセクションでは、Schreier-Simsアルゴリズムの基本概念を紹介する。この後、**拡張Schreier-Simsアルゴリズムで**Schreier-Simsアルゴリズムを使用する。この最後のアルゴリズムは、**拡張メンバシップ検査**アルゴリズムを使用するためのデータ構造を作成する。この拡張メンバシップ検定アルゴリズムは、与えられた順列がSym(n)の与えられた部分集合T中の単語として書けるかどうかを決定し、もし書けるならば、その順列のT中の単語を出力する。

ここでは群$G\subset \mathrm{Sym}(n)$を考える。

定義6.1. G*の*基底とは、数列$B = (\beta_1 , - - , \beta_m)$ *であり、*β_i は$\{1, . . \mathrm{n}\}$*の部分集合であり*、B*を*固定するG*の*唯一の要素は恒等式 *である。*

$B = (\beta_1 , - - , \beta_m)$ をベースとする。$G^{[i]} := G_{(\beta_1,\cdots,\beta_{i-1})}$ をポイントワイズとする。

の安定化子 $(\beta_1 , - - , \beta_{i-1})$。ここで$B$*は*部分群鎖を定義する：

$$g = g^{[1]} \geq g^{[2]} \geq - - - \geq g^{[m]} \geq g^{[m+1]} = 1.$$

定義 6.2.基底Bは、上で定義した部分群連鎖のすべての部分群が、前任者の適切な部分群である場合、**非冗長と**呼ばれる。

定義6.3.G の基底 $B = (\beta_1, , \beta_m)$ に対する**強い生成**集合とは、$1 \geq i \geq m + 1$ に対して以下のような G の生成集合 S である：

$$S \cap G^{[i]}\rangle = G^{[i]}。 \tag{1}$$

定義6.4. HをGの部分群とする。HをGの部分群とする。$G \bmod H$の（左）**横線**Rは、Hの各左余集合のちょうど1つの要素を含むGの部分集合である。

定義6.5.ここでTは有向グラフであり、その下にある無向グラフはサイクルを含まず、他の各頂点に到達できる頂点が存在する。この頂点は一意であり、この頂点を木の根と呼ぶ。また、fはTの辺の集合からSへの写像であり、Sを有向根付きラベル木のラベル集合と呼ぶ。

$_{1m}$iを1≦i≦mの自然数とし、sを$G_{(\beta 1,\ldots,\beta i-1)}$の$s$とし、vを$G_{(\beta 1,\ldots,\beta i-1)}$の$G_{(\beta 1,\ldots,\beta i)}$の左コセットとする。$G_{(\beta 1,\ldots,\beta i-1)}$の左余集合に対する$G_{(\beta 1,\ldots,\beta i-1)}$の自然作用は、$s(v) = sv$によって定義される。

定義6.6. Sym(n)の部分集合Tが与えられたとき、**シュライア木データ構造**Δ Tの△は次のようになる。

- (T)のベース$B = (\beta_1, \ldots, \beta_m)$、および
- i∈｛1〜-m｝のm本の有向根付きラベル木（T_i、S_i、f_i

）。

- S_i は$G_{(\beta 1,\ldots,\beta i-1)}$の部分集合である、
- $_i$の$G_{(\beta 1,\ldots,\beta i)}$の左余集合を頂点とする有向木である。$G_{(\beta_1,\ldots,\beta_i-1)}$。
- 頂点$G_{(\beta 1,\ldots,\beta i)}$は木の根である、

- T_i のすべての辺 e が頂点 γ から頂点 δ を指し、$f(e) = s$ である。
  は$s(\gamma) = \delta$という性質を持つ。

シュライアーツリーデータ構造における有向根のラベル付きツリーは、次のように呼ばれる。

シュライヤーの木。

定理6.1. *GをSym(n)の部分群とし、$B = (\beta_1, \beta_2, \dots, \beta_m)$ をGの基底とする。iを$1 \leqq i \leqq m$の自然数とする。Cを$G_{(\beta_1,\dots,\beta_{i-1})}$ における$G_{(\beta_1,\dots,\beta_i)}$ の左余集合の集合とする。A を $G_{(\beta_1,\dots,\beta_{i-1})}$ におけるβ_i の軌道とする。Cの各gとg_1 がgの中にあるとき、$f(g) = g_1(\beta_i)$ によって写像fを定義する。そして、fはCからAへのよく定義された両対称写像fである。*

*証明*まず、fがよく定義されていることを示す。g_2 もgの中にあるとする。すると、$g h_{11} = g_2$ となるようなh_1 が$G_{(\beta_1,\dots,\beta_i)}$ の中に存在する。

$$g_2(\beta_i) = g h_{11}(\beta_i) = g_1(h_1(\beta_i)) = g_1(\beta_i).$$

$G_{(\beta_1,\dots,\beta_{i-1})}$における$\beta_i$ の軌道内のすべての要素$\alpha \in A$に対して、$G_{(\beta_1,\dots,\beta_{i-1})}$において、$g_3(\beta_i) = \alpha$となるような$g_3$ が存在する。定義により、fはg_3 を含む$G_{(\beta_1,\dots,\beta_{i-1})}$ の$G_{(\beta_1,\dots,\beta_i)}$ のcosetをαに写像するので、fは射影である。

g' を、$f(g') = f(g) = g_1(\beta_i)$ となるような C にあるとする。$g1'$ を g' とする.

$g1'(\beta_i) = g_1(\beta_i)$ したがって、これらは$G_{(\beta_1,\dots,\beta_i)}$ の$G_{(\beta_1,\dots,\beta_{i-1})}$ の同じ左余集合にある。したがって、$g' = g$である。これはfが射影であることを示している。

□

fは両対称であると結論づけられる。

*備考*6.1.上記の定理により、我々は$G_{(\beta_1,\dots,\beta_i)}$ のコセットを以下のように識別する。

$G_{(\beta_1,\dots,\beta_{i-1})}$の下での$\beta_i$ の軌道の要素で$G_{(\beta_1,\dots,\beta_{i-1})}$。

レンマ6.2. *TをSym(n)の部分集合、ΔをSchreier木データ構造とする。*

T_i /はT_i。 $/^{Qm}$ /T_i | = /$\langle T \rangle$ | とする。 i=1

証明。T_i の頂点の数は、$G_{(\beta_1,\dots,\beta_{i-1})}$における$G_{(\beta_1,\dots,\beta_i)}$ のコセットの数に等しい。これは$G_{(\beta_1,\dots,\beta_{i-1})}$における$G_{(\beta_1,\dots,\beta_i)}$のインデックスに等しい。したがって、反復的に

$$|T\langle T\rangle| = |T_1| \cdot |T_{(\beta_1)}| = |T_1| \cdot |T_2| \cdot |G_{(\beta_1,\beta_2)}| = \dots = \prod_{i=1}^{m} |T_i|.$$

□

定義6.7.ベース$B = (\beta_1 , \ldots , \beta_m)$ を持つシュライア木データ構造Δに対して、ΔのカーディナリティはBのβ_i の最大のカーディナリティである。

$B = (\beta_1 , \beta_2 , \ldots , \beta_m)$ をGの基底とする。Bに対するGの強い生成集合Sがあれば、$G^{[i]} \beta_i$ の軌道と、$G^{[i]} \bmod G^{[i+1]}$ の横軸を以下のように計算できる。データはシュライアー木構造で保存する。S_i を $S\ G^{[i]}$ とする。i が$1\ i\ m$ のとき、計算は以下のようになる:

1. T_i が空の状態から始め、頂点 β_i を追加する。

2. T_i に新しく追加された各頂点 t と、S_i の各 s について、$s(t)$ を計算し、$s(t)$ が T_i にない場合、それを T_i に追加し、s とラベル付けされた t から $s(t)$ への辺を追加する。

3. 新しい頂点が追加されていなければ、やめる。そうでなければステップ2を繰り返す。

各頂点γは、β_i をγに移動させる$G^{[i]}$ の要素からなる$G^{[i]}$ の$G^{[i+1]}$ の左コセットに対応することに注意する。ここで、γをT_i のそのような頂点とする。 β_i からγへの一意なパスが存在する。($s_1 , s_2 , \ldots , s_t$) がβ_i から始まるこのパスに沿ったラベルであるならば、 $s\ s_{tt-1}\ s_1$ は$G^{[i]}$ の要素であり、シュライアー木の定義によってβ_i をγに移動させる。この要素を、γに対応する左cosetに含まれるR_i のユニークな要素とすることで、$G^{[i]}$ mod $G^{[i+1]}$ の横軸R_i を構成する。これをすべての頂点について行うことで、完全な横軸R_i が得られる。

*アルゴリズム*6.1.上で作成したようなシュライア木データ構造Δを用いると、G中のすべてのgを$g = r\ r_{12}\ r_m$ with r_i in R_i のように書くことができる。この手順はΔ**をふるいにかけると**呼ばれ、以下のように行われる:

1. $i = 1$とし、$g_1 = g$とする。

2. β_i から g_i (β_i) までの一意なパスが存在する。($s_1 , s_2 , \ldots , s_t$) を β_i から始まるこのパスに沿ったラベルとすると、$r_i = s\ s_{tt-1}$ - - s とする。 $_1$

3. i が m であれば停止する。そうでなければ、$g_{i+1} = r\ g^{-1}{}_i$ とする。g_{i+1} は $G_{(\beta\ 1,\ldots,\beta i)}$.$i$に1を加え、ステップ2を繰り返す。

ここで $r\ g^{-1}{}_m$ は $G_{(\beta,\ldots,\beta)m}$ = {1} の中にあり、したがって $r\ r_{12}$ - - $r_m = g$ である。なぜこのアルゴリズムが機能するのか、次のことを考えてみよう。要素r_1 はβ_1 を$g(\beta_1)$に移動させ、要素r_2 はβ_2 を$g_2(\beta_2) = r1\text{-}1g(\beta_2)$に移動させるが、$\beta_1$ は移動させない。したがって、要素$r\ r_{12}$ はβ_1 とβ_2 をそれぞれ$g(\beta_1)$と$g(\beta_2)$に移動させる。

*備考*6.2.ふるいにかける過程で、1≦i≦mに対してr_i を構築する。r_i は、シュライア木データ構造のシュライア木のエッジのラベルを乗算す

ることで構築する。

*備考*6.3.Sym(n)のgがGのメンバーかどうかを調べるために、ふるいを使うこともできる。要素gがGに含まれないのは、あるiについて$g_i(\beta_i)$がT_iの頂点でないか、または$g\,r\,r^{-1}{}_{12}\,r_m$が恒等式でない場合のみである。

定義6.8. Sym(n)中のgについて、計算可能で、計算可能なインデックスの中で最も高いインデックスiを持つg_i をgの**シフティーと呼ぶ**。

Schreier-Simsアルゴリズムには、以下の2つのレンマが必要である。

レンマ6.3。 *$H\ G = S$とし、RをG mod Hの左横線とする。*

$$T = \{\overline{(sr)}^{-1}\, sr | s \in S,\ r \in R\}.$$

Tの要素はHのシュライヤー生成子と呼ばれる。

*証明しよう。*定義により、Tの要素はHの中にあるので、集合$T \cup T^{-1}$がHを生成することを示せば十分である。 $T^{-1} = \{(sr)^{-1}\ sr|s \in S^{-1}$, r∈R} に注意。$h\ H$を任意のものとする。$_{01}H$は$G$であるから、$h$は$h = s$ $s_{kk-1}\ s_1$ with $s_i\ S\ S^{-1}$ の形で書くことができる。h_kを定義する。

$$h_j = s_k - - s\ s\ r_{j+2j+1j+1j}\ \ t- - t\ t_{21}$$

$t_i\ T\ T^{-1}$, $r_{j+1}\ R$ and $h_j = h$. h_0 を $s_k\ s\ s\ r_{211}\ r_1 =$ とする。

1. 再帰的に、h_j がすでに定義されている場合、t_{j+1} を $(s\ r)\ s_{j+1j+1}{}^{-1}{}_{j+1j+1}$ rとし、r_{j+2} を $s\ r_{j+1j+1}$ とする。明らかに $h_{j+1} = h_j = h$ であり、必要な形式を持つ。

$h = h_k = r\ t_{k+1k} - - t\ t_{21}$ が成り立つ。$h \in H$ および $t_k - - t\ t_{21} \in \langle T \rangle \leq H$ なので、$r_{k+1} \in H \cap R = \{1\}$でなければならない。したがって、$h \in \langle T \rangle$となる。□

定理6.4.$(\beta_1, \ldots, \beta_k)$ *を*$\{1, \ldots, n\}$*の部分集合の列とし、*G*を*$Sym(n)$*の部分群とする。*n} の部分集合列で*あり、*G *は*Sym(n) *の部分群である。*$_{j+11k+1}$1 ≤ j ≤ k + 1 *に対して、安定体*$G_{(\beta\ 1,\ldots,\beta j-1)}$*の部分集合で、すべての*$j \leq k$ *に対して* $\langle_j \rangle \geq \langle \rangle$ *が成り立つものを*S_j *とする。*

$$_j \langle \rangle\ j_\beta = \langle S_{j+1} \rangle。 \tag{2}$$

すべての$1 \leq j \leq k$*に対して、*$B = (\beta_1, \ldots, \beta_k)$ *は*G*の基底であり、*$S = \bigcup_{1 \leq j \leq k} S_j$ *は*B*に対する*G*の強生成集合である。*

*証明。*帰納的仮説は、$S^* = \cup\ S_{2 \leq j \leq kj}$ は $\langle S_2 \rangle$ の強い生成集合であり、基底$B^* = (\beta_2, \ldots, \beta_k)$ に相対する。$G^{[i]}$ *を* $G_{(\beta\ 1,\ldots,\beta i-1)}$ とする。定義により、$i = 1$のとき(1)が成立する。2≦i≦k＋1について(1)が成り立つことを確認しなければならない。$i = 2$の場合、$j = 1$で(2)を適用すると、$G_\beta 1 = \langle S_2 \rangle$ ↪S∩ $G_{\beta 1)}$が得られるので、(1)が成り立つ。逆包含は明らかである。$i > 2$の場合、$S^* \cap G_{(\beta\ 1,\ldots,\beta i-1)}$ が以下を生成することから、(1) が成り立つ。

$\langle S2 \rangle_{(\beta_2,\ldots,\beta_{i-1})}$帰納的仮説により、$G^{[i]} \geqq$ ↪S_27E8↩S∩ $G_{(\beta_1,\ldots,\beta_{i-1})} \rangle \geq \langle S* \cap G_{(\beta_1,\ldots,\beta_{i-1})} \rangle \rangle = \langle_{(\beta_2,\ldots,\beta_{i-1})} = (G\beta_1)_{(\beta_2,\ldots,\beta_{i-1})} = G^{[i]}$。 □

7 Schreier-Sims アルゴリズム

このセクションでは、Schreier-Simsアルゴリズムについて説明する。

このアルゴリズムの入力は、あるnに対するSym(n)の部分集合Tである。

レンマ6.3を使えば、各$G^{[i+1]}$ $G^{[i]}$ に対して生成 子の集合を作ることができ、Bに対するGの強い生成集合を作ることができる。したがって、新しい生成子を追加するときは、まずこの生成子が冗長かどうかをテストする。これは以下に説明するようにふるいにかけることで行うことができる。ある群でふるいにかけるには、その群の強い生成集合が必要であることに注意。

Sym(n)の部分集合Tが与えられる。Gを$\langle T \rangle$とする。$_i$我々は、Gに対する冗長でない基底の既知の要素のリスト$B = (\beta_1 , \ldots , \beta_{m'})$を保持する。また、リスト$(S_1 , S_2 , \ldots , S_i , \ldots S_{m'})$を保持する。

$1 \leqq i \leqq m'$ ）に対するスタビライザー$G_{(\beta_1,\ldots,\beta_{i-1})}$）のジェネレーター・セットに対して、我々は常に$1 \leqq i \leqq m'$ を維持する：

$$_{i+1} \langle \rangle \leq \langle_j \rangle_\beta \, \mathrm{i} \leq \langle_j \rangle$$

$k < j$であるすべてのjについて式(2)が成り立つ場合、データ構造は$\leq$ *レベルk以下で最新*であると言う。 m。′

下図はその状況を示したものである；

$$
\begin{array}{ccc}
\langle S_1 \rangle & \subset & G \\
\cup & & \cup \\
\langle S_2 \rangle & \subset & G_{(\beta_1)} \\
\cup & & \cup \\
\vdots & \vdots & \vdots \\
\langle S_j \rangle & \subset & G_{(\beta_1,\ldots,\beta_{j-1})} = G^{[j]} \\
\cup & & \cup \\
\langle S_{j+1} \rangle & \subset & G_{(\beta_1,\ldots,\beta_j)} = G^{[j+1]}
\end{array}
$$

*備考*7.1. アルゴリズム中に新しい基点を選択する。この節では、基点の選び方について詳しくは述べない。しかし、結果として得られるシュライア木データ構造がカーディナリティ$r = 1$になるような基点を選ぶことができることに注意する。

*備考*7.2. このアルゴリズムで構築するベースは冗長ではない。

次にそのアルゴリズムを説明する。

Gを$\langle T \rangle$とする。我々は、Gの冗長でない基底の既知の要素のリスト$B = (\beta_1, \ldots, \beta_{m'})$と、$1 \leq i \leq m$のスタビライザー$G_{(\beta_1,\ldots,\beta_{i-1})}$の生成集合の近似$S_i$を保持する′。我々は、常に、すべての$i$について、$S_i$ S_{i+1}を持つという特性を保持する。$j < i$ m' であるすべてのiについて式(2)が成り立つ場合、データ構造は*レベルj以下で*最新であると言う。

Schreier-Simsアルゴリズムを以下のように実行する：

1. $S_1 = T$とし、$j \geq 2$については$S_j = \varnothing$とする。Tの生成子によって移動される、最大rのカーディナリティを持つ$\{1, \ldots, n\}$の部分集合β_1を選択することによってアルゴリズムを開始し、$m' = 1$と設定する。これで、レベル$j = 1$以下のデータ構造は最新となる。

2. データ構造がレベルj以下で最新である場合、$\langle S_j \rangle \bmod \langle_j \rangle_\beta \, \mathrm{j}$ の

シュライアツリー(T_j , S_j , f_j)を計算する。

3. これは、シュライア ツリー（T_j , S_j , f_j ）によって符号化された横軸の要素と、 S_j の要素を用いて得られるシュライア生成子（レンマ6.3参照）をふるい分け、レンマ6.3を適用して $S_{j\beta_j}$ の生成子を得ることによって行うことができる。グループS_{j+1} でふるいにかける。これは、データ構造がレベルj以下で最新であり、6.4によって ⟨S$_{j+1}$ ↪Pe_27E9 ⟩ の強い生成集合を持つため可能である。

4. 私たちは今、2つのケースを見極めようとしている。
   - もしjに対して式(2)が成り立つなら、データ構造はレベルj - 1以下が最新である。
   - そうでない場合は、自明でないシフティを持つシュライア生成子sが存在する。$_{j+1}$もし$j = m'$ならば、β_{j+1}を、sによって移動される最大rのカーディナリティを持つ$1, , n$の新しい部分集合として選び、m'を1つ増やす。これでデータ構造はレベル$j + 1$以下の最新のものになった。
5. レベル0以下のデータ構造が最新であれば終了。Lemma 6.4は正しさを暗示している。そうでなければステップ2に進む。

*備考*7.3.本アルゴリズムの実装では、ステップ2でシュライヤー木全体(T_j , S_j , f_j)を再計算しない。その代わりに、すでに計算されたシュライヤー木を保存し、S_jの新しい要素に対して、この要素を前のシュライヤー木の各頂点に適用する。これにより、更新されたシュライア木(T_j , S_j , f_j)が得られる。

8 Schreier-Sims アルゴリズムの複雑さ

ここで、上述したSchreier-Simsアルゴリズムの複雑性を解析する。この解析は[3]に類似しているが、我々は基底のカーディナリティrを導入している。

*表記*本稿では、logは2を底とする対数を表す。

定理8.1. *入力Tと自然数$r \geq 1$をとるSchreier-Simsアルゴリズムは、$O(n\binom{n}{r}^2 \log^3 |G|+n\binom{n}{r}^2 |T| \log |G|)$時間を用いて、最大$r$のカーディナリティのSchreier木データ構造$\Delta$を$O(n \log^2 |G|+\binom{n}{r} \log |G|+|T| n)$時間で構築する。メモリだ。*

証明。 基底の長さは最大でも対数｜G｜である。上記のようにステップ2のシュライア木(T_1 , S_1 , f_1)を計算する際、Tの各要素をT_1の最大で$\binom{n}{r}$個の頂点に適用する。 これには$O(\binom{n}{r}|T|)$がかかる。を持

つ基底の固定β_k 。

$k > 1$, グループ $\langle S_k \rangle$ は要素を追加するたびに増えるので、セットS_k はアルゴリズム中に最大でも$\log |G|$回変化する。の変更後

S_k ⟨では、シュライア木(T_k, S_k, f_k)を更新しなければならない。$\binom{n}{r} T_k$

。 ⟨S$_k$ ↪Pe_27E9 ⟩に要素sが追加されたとき、上記のようにステップ2のシュライア ツリーを更新するには、sの下でのT_k の各頂点のイメージを1回だけ計算する必要がある。並べ替えの下での点のイメージの計算は

$O(1)$.したがって、この計算には $O(\binom{n}{r}|B|) = O(\binom{n}{r} \log |G|)$ がかかる。したがって、すべての

上記のステップ2と同様に、すべてのシュライアツリーを構築するために、基点が必要である。

$O(\binom{n}{r}|B| \log |G| + \binom{n}{r}|T|)$である。

$$O(\binom{n}{r} \log^2 |G| + \binom{n}{r}|T|)。 \tag{3}$$

ここで、上記のようにステップ3のすべてのシュライヤー世代をふるいにかけるのにかかる時間を見積もる。横線R_k の各要素は、次の要素と組み合わされなければならない。

のS_k。したがって、シュライアー生成子の総数は $\sum_k |R_k||S_k|$ となり、ここで $|R_k|$ と $|S_k|$ はアルゴリズム終了後に考慮される。ここで、 /R /と /$S^n{}_k$ |はアルゴリズム終了後に考える。$S_1 = T$であり、$k > 1$の場合、次のようになる。

$|S_k| = O(\log|G|)$。したがって

$$\sum_k |R_k||S_k| = O(\binom{n}{r}\log^2|G| + \binom{n}{r}|T|)。$$

シュライヤー木からR_k の要素を取り出すには、シュライヤー木のパスに沿ってすべてのラベルを乗算しなければならない。これは、それぞれ$O(n)$を要する最大$\binom{n}{r}$の並べ替え乗算を意味する。我々はこれらの要素のうち、最大で$\log G$を取り出す必要があるので、1回のふるいにかかるコストは次のようになる。

$$O(n\binom{n}{r}\log|G|).$$

すべてのシュライヤー・ジェネレーターをふるい分けるための総コストは、1回のふるい分けのコストにシュライヤー・ジェネレーターの数を掛けたものである。これは

$$O(n\binom{n}{r}^2\log^3|G| + n\binom{n}{r}^2|T|\log|G|)。 \qquad (4)$$

(3)と(4)により、アルゴリズムの総時間コストは次のようになる。

$$O(n\binom{n}{r}^2\log^3|G| + n\binom{n}{r}^2|T|\log|G|)。$$

我々は$\sum$k | S /$_k$ |強生成子を保存しなければならないが、これは最大でも$O(\log^2|G|)$である、

したがって、$O(n\log|G|)$のメモリを必要とする。 シュライヤー木を保存するには $O(\binom{n}{r}\log|G|)$ メモリ。基底を格納するのにかかるmrは$O(r\log|G|)$であるが、$r \leq n$なので、これは上記の$O(n\log^2|G|)$に支配される。初期ジェネレータを格納するために使用されるT n個のメモリをカウントすると、以下の総メモリコストがあると結論付けられます。

$$O(n\log^2|G| + \binom{n}{r}\log|G| + |T|n).$$

□

9　拡張Schreier-Simsアルゴリズムと拡張メンバシップテスト アルゴリズム

このセクションでは、順列を入力として与えられた拡張メンバーチップ検査アルゴリズムが、この順列に対する単語を出力するデータ構造を作成する、Schreier-Simsアルゴリズムの適応について説明する。

定義9.1.TをSym(n)の部分集合とする。ベース$B = (\beta_1, \ldots, \beta_m)$ を持つTのシュライア木データ構造Δが与えられたとする。に対して3種類の**シュライア ポインタを**定義する：

- タイプ1：シーケンスp＝（p_1、q_1、1、$\tilde{p}$）ここで、p_1はTにあり、$q_1 \in$｛1、-1｝および $\tilde{p} = p_1^{q_1}$.

- タイプ2：シーケンス$p = (p_1 , q_1 , \dots , p_c , q_c , i, p\sim)$で、$2 \leqq i \leqq m$であり、$p_k$が別のシュライアポインタであるようなシーケンス、
  $p'_k = (p'_1, q_1', \dots , p'_{c'}, q_{c'}', i', j')$ または
  $p'_k = (p'_1, q_1', \dots , p'_{c'}, q_{c'}', i', p\sim')$, with $i' < i$ and $j' \in G^{[i]} \beta_i$ and q_k is in.
  すべてのkに対して{-1, 1}であり、$p\sim = \prod_{k=1}^{c}$である。$p\sim_k^{q_k}$。

- タイプ3：どちらか

  – シーケンス $p = (p_1 , q_1 , i, j)$ ここで、p_1 は T にあり、$q_1 \in \{1, -1\}$, $1 \leq i \leq m$ である。
  およびj（G）$^{[i]} \beta_i$ または

  – シーケンス $p = (p_1 , q_1 , \dots , p_c , q_c , i, j)$ で、$2 \leq i \leq m$ であり、p_kが別のシュライアーポインタであるようなシーケンス。. . , $p'_{c'}, q_{c'}', i', j'$）、または$p'_k = (p'_1, q_1', \dots, p'_{c'}, q_{c'}', i')$、$i' < i$ で、$j \in G^{[i]} \beta_i$ のいずれかである。また、q_k はすべてのkについて{-1, 1}にある。

タイプ3のシュライアー・ポインターでは、$p\sim = \prod_{k=1}^{c}$と定義する。 $\tilde{p}_k^{q_k}$.

タイプ1のシュライヤーポインターとタイプ3のシュライヤーポインターは長さ1と呼ばれ、タイプ2のシュライヤーポインターとタイプ3の シュライヤーポインターは長さcと呼ばれる。シュライアー・ポインター$p = (p_1 , q_1 , \dots , p_c , q_c , i, j)$、または$p = (p_1 , q_1 , \dots , p_c , q_c , i, p\sim)$は$i$番目のシュライアー木にあるという。

定義 9.2.シュライア木データ構造Δ、**ポインタ構造**

からΔへの写像は、以下のようなシュライアーポインタの集合Pである。

$$\{p \in P \mid p\text{はタイプ3のシュライアポインタ}\} \to \{(i, j) : 1 \leqq i \leqq m, j \in G^{[i]} \beta_i , j \neq \beta_i\}$$

$$(p_1 , q_1 , \dots , p_c , q_c , i, j) \mapsto (i, j)$$

は両対称であり、P内のすべてのpについ て、pが$\bar{T}_i$のルートから頂点jに至る経路のユニークな最後の辺のラベルに等しいことがわかる。シュライアーポインタpはこのユニークな最後の辺に対応するという。

拡張Schreier-Simsアルゴリズムは、Sym(n)の部分集合Tと整数rを

入力とし、最大rのカーディナ リティのSchreier木データ構造Δと、Δに対するポインタ構造P の両方を出力する。このアルゴリズムはSchreier-Simsアルゴリズムを実行する。以下のシュライアーポインタがPに追加される：

- 初期化時に、Pにタイプ1のシュライアーポインタ$(t, 1, 1, t)$を追加する。$t \in T$。
- Schreier-Simsアルゴリズムのステップ4では、新しいSchreier生成子sがS_{j+1} に追加された場合、タイプ2のSchreierポインタをPに追加する。$^{-1''}{}_{jj}$追加されたシュライア ポインタ$p = (p_1 , q_1 , \ldots , p_c , q_c , i, p\sim)$は以下のように構成される。

– j番目のシュライアツリーには、ルートから次のようなユニークなパスがある。$s'r(\beta_i)$ は、エッジに対応するシュライアーポインタ $(a_1 , \ldots , a_n)$ を持つ。$(s'r)^{-1}$ となるような $\tilde{a}_n \cdots \tilde{a}_1 = s'r$となる $\tilde{a}_1^{-1} \cdots \tilde{a}_n^{-1} =$ このパスの。 。

– j番目のシュライヤー木には、$\sim$b = s' となるシュライヤー・ポインターbが存在する。

– ルートから$r(\beta_i)$までのj番目のシュライヤー木には、$c\sim_m \sim$ $c\sim_1$ ＝rとなるような、このパスのエッジに対応するシュライヤーポインタ$(c_1, \ldots, c_m)$を持つユニークなパスが存在する。

– に追加したシュライアー・ポインターを設定する。

$$p = (a_1 , -\underline{1,} \ldots , a_n , -1, b, 1, c_m , 1, \ldots c_1 , 1, j + 1, p\sim)。$$

p˜ = $(s'r)$ $s^{-1'}$ r = s が成り立つ。

- Schreier-Simsアルゴリズムのステップ2では、i番目のSchreier木を更新する。S_i に新たに追加された各シュライヤー生成子s に対して、この s を既存のシュライヤー木の各頂点に適用する。sを適用した結果、更新されたSchreier木に新しい頂点v' が生じる各頂点vに対して、拡張Schreier-Simsアルゴリズムは、Pにタイプ3のSchreierポインタpを追加する。

  $i = 1$ ならば、 s はa t に等しい。　T.追加したシュライヤー・ポインターを $(t, 1, 1, v')$ にセットする。

  もしi2 の場合、i番目のシュライア木に$p\sim$＝sのタイプ2のシュライア・ポインタp＝ $(p_1 , q_1 , \ldots , p_c , q_c , i, p\sim)$ を追加したことになる。シュライアー・ポインター$p' = (p_1 , q_1 , \ldots , p_c , q_c , i, v')$ を追加する。

タイプ2のシュライアー・ポインターには強生成子が含まれているので、実装では強生成子のリストの代わりにタイプ2のシュライアー・ポインターのリストを保持することができる。

Sym(n)の部分集合Tに対するシュライア木データ構造Δと、Δに対するポインタ構造Pに対する**拡張メンバシップ検査アルゴリズムは、**順列g Sym(n)を入力とする。もし s が T にあれば、T の単語を g.これはΔを通してgをふるいにかけることによって行われる。Remark6.2

により、ΔのSchreier木における辺のラベルを掛け合わせることで、 篩い分けの過程でgを書く。定義によるポインタ構造Pは、Δのシュライア木の各辺のラベルごとに単語をエンコードする。

i番目のシュライヤー木をふるいにかけるとき、β_i から $g_i(\beta_i)$ までの、辺 $(e_1, \ldots, e_n)$ を持つ一意なパスが存在する。まず、e_n に対応するシュライヤー・ポインターでエンコードされた単語を出力し、次に e_{n-1} に対応するシュライヤー・ポインターでエンコードされた単語を出力する。アルゴリズム6.1は、結果として得られる出力がT中のgに対応する単語であることを示している。

10　出力ワード 長さ

*表記*シュライア木T_i が与えられたとき、シュライア木の高さを、根から木の頂点に向かう最長の経路と定義する。例えば、頂点が1つの木は高さ0である。木の高さを$h(T_i)$とする。$h(T_0) = 0$ と定義する。

このセクションでは、拡張メンバシップ・アルゴリズムの出力である単語の上限長を証明する。以下のレンマを用いる。

定理10.1. *拡張Schreier-Simsアルゴリズムによって構築されたポインタ構造において、i番目のSchreier木に対するSchreierポインタの長さは最大で*

$$2 - h(T_{i-1}) + 1$$

*i≧2ではちょうど*1、*i*=1では*ちょうど*1である。

証明。 i*番目の*木のシュライヤー・ポインターは3つの部分からなる。$(sr)^{-1}$、*r*に対応する部分の長さは、最大でもT_{i-1}の最長経路に等しい、つまり最大でも$h(T_{i-1})$である。*s*に対応する部分の長さは1である。したがって、シュライア ポインターの長さは最大でも2 $h(T_{i-1}) + 1$である。最初のシュライヤー木の各シュライヤー・ポインターの長さは、定義により1である。

定理10.2. *入力T* Sym(*n*)、*s T* =*G*、*拡張Schreier-Simsアルゴリズムによって構築されたシュライア ツリーデータ構造*Δと Δに対する*ポインタ構造Pが与えられた。このとき、拡張メンバシップ検査アルゴリズムは、T中の文字を持つsに対して、｜G｜2文字より長くない単語を出力する。*

証明。 使用する基底の長さをmとする。レンマ10.1により、*i番目の*シュライア木におけるシュライアポインターの最大長は

$$2 - h(T_{i-1}) + 1.$$

拡張メンバシップ検査アルゴリズムでは、*s*をΔの中でふるいにかける。*i番目の*Schreier木をふるいにかけるとき、*s*の単語を出力するために最大$h(T_i)$のSchreierポインタが使われる。Schreierポインタは(*i* 1)番目のSchreier木のポインタだけを指す。シュライヤー・ポインターが最初のシュライヤー木にある場合、文字を出力する。従って、i*番目の*シュライヤー木の辺に対応するシュライヤー・ポインターを明示的に書くには、最大で

$$\sum_{k=1}^{iY-1} (2 - h(T_k) + 1)$$

文字が出力される。したがって、i *番目の*シュライアツリーをふるいにかける間に、 最大で

$$h(T)_i \prod_{k=1}^{i-1} (2 - h(T_k) + 1)$$

文字が出力される。ふるい分けプロセス全体では、最大でも

$$\sum_{i=1}^{m} h(T)_i \prod_{k=1}^{i-1} (2 - h(T_k) + 1)$$

文字が出力される。これは

$$\sum_{i=1}^{m} 2^{i-1} - \prod_{k=1}^{i} (h(T_k) + 1) .$$

ここでQ $h(T_k) + 1$は、最大でもT_kの頂点の数である。 $_i |T_i| = |G|$ である。であるから、これはせいぜい

$$\sum_{i=1}^{m} 2^{i-1} \cdot |g| < 2^m |g|.$$

したがって、私たちが出力する単語の長さは最大でもlog Gである。

$$|G|^2.$$

□

11 複雑さ 分析

ここで、拡張Schreier-Simsアルゴリズムと拡張メンバシップテストアルゴリズムの複雑さを計算する。

Sym(n)に順列を格納するコストはnである。

定理11.1. Sym(n)*の部分集合*T*と自然数*$r \geq 1$が*与えられたとき、拡張Schreier-Sims アルゴリズムは、*T*に対するSchreier木データ構造*Δ*と、*Δに対する*ポインタ構造P を出力する。*

$$O(\tbinom{n}{r} \log^2 |G| + \tbinom{n}{r}^2 \log |G|) + |T|n)$$

そして

$$O(n\tbinom{n}{r}^2 \log^3 |G| + n\tbinom{n}{r}^2 |T| \log |G|)$$

時間である。

*証明。*定理8.1により、Schreier-Simsアルゴリズムが必要とするメモリは以下の通りである。

$$O(n \log^2 |G| + \tbinom{n}{r} \log |G| + |T|n).$$

拡張Schreier-Simsアルゴリズムでは、各Schreier木の各辺に対してSchreierポインタを格納する。また、各強生成子に対してもシュライヤー・ポインターを格納する。

Lemma 10.1により、i*番目の*シュライア木におけるシュライア・ポインタ$_r$の長さは最大でも $2 \cdot h(T_{i-1}) + 1$であることがわかる。シュラ

イア木の高さはせいぜい$\binom{n}{r}$である。したがって、シュライヤー・ポインターの長さは最大でも$2 \cdot \binom{n}{r} + 1$である。$i \geq 2$

シュライアポインタを格納するために必要なメモリは、シュライアポインタの長さに等しい。$i = 1$の場合、シュライヤーポインターを格納するコストはSym(n)の順列を格納するコストに等しく、$n \leq \binom{n}{r} < 2 \cdot \binom{n}{r} + 1$となる。

シュライア木は最大で$\binom{n}{r}$個の頂点を含む。シュライア木は最大でも$\log |G|$個存在する。定理8.1の証明で見たように、我々は$O(\log^2 |G|)$の強い生成子を保存する。したがって、すべてのシュライヤー・ポインターを保存するコストは最大で

$$O\left(\left(2 \cdot \binom{n}{r} + 1\right)\left(\log^2 |G| + \binom{n}{r} \log |G|\right)\right) = O\left(\binom{n}{r} \log^2 |G| + \binom{n}{r}^2 \log |G|\right)$$

総メモリ量は以下の通りである。

$$O(n\log^2|G| + \binom{n}{r}\log|G| + |T|n + \binom{n}{r}\log^2|G| + \binom{n}{r}^2\log|G|)$$

それは

$$O(\binom{n}{r}\log^2|G| + \binom{n}{r}^2\log|G| + |T|n).$$

定理8.1により、Schreier-Simsアルゴリズムは以下の所要時間を持つ。

$$O(n\binom{n}{r}^2\log^3|G| + n\binom{n}{r}^2|T|\log|G|)。$$

シュライヤー・ポインターの追加には、そのシュライヤー・ポインターの長さに等しい時間が必要である。したがって、上記と同様に、シュライアーポインターの追加には

$$O(\binom{n}{r}\log^2|G| + \binom{n}{r}^2\log|G|)$$

の合計時間が必要だと結論づける。

$$O(n\binom{n}{r}^2\log^3|G| + n\binom{n}{r}^2|T|\log|G| + \binom{n}{r}\log^2|G| + \binom{n}{r}^2\log|G|)$$

に等しい。

$$O(n\binom{n}{r}^2\log^3|G| + n\binom{n}{r}^2|T|\log|G|)。$$

□

定理11.2. *Tを*Sym(n)*の部分集合とし、ΔとPを、拡張Schreier-Simsアルゴリズムによって作成された、Tに対するSchreier木データ構造と*Δに対する*ポインタ構造とする。拡張メンバシップアルゴリズムは、入力s*∈Sym(n)が*与えられたとき、T中のsに対す る単語を出力する。*

$$O(n\cdot|G|^2 + n\binom{n}{r}\log|G|)$$

時間である。

*証明*定理8.1の証明で見たように、Schreier-Simsアルゴリズムにおける1つの篩のコストは$O(n\binom{n}{r}\log|G|)$である。

拡張メンバーシップ検査アルゴリズムもポインタ構造を使用する。メ

ンバシップ検定で使われるすべての辺のラベルに対して、その辺に対応するシュライアーポインタを明示的に書かなければならない。レンマ10.1により、i *番目の*シュライヤー木におけるポインターは、長さが最大で

$$2 - h(T_{i-1}) + 1.$$

このようなポインターはi *番目の*シュライヤー木のポインターだけを指す。最初のSchreier木に明示的にSchreierポインターを書くにはn *の*コストがかかる。

$$n \cdot \sum_{k=1}^{i-1} (2 \cdot h(T_k) + 1)$$

時間である。したがって、i*番目の*シュライアツリーをふるいにかけるのにかかる時間は、最大で

$$n \cdot h(T)_i \prod_{k=1}^{i-1} (2 \cdot h(T_k) + 1)。$$

すべてのシュライアーの木をふるいにかけるには、最大でも次のような時間がかかる。

$$n \cdot \sum_{i=1}^{m} h(T)_i \prod_{k=1}^{i-1} (2 \cdot h(T_k) + 1) .$$

よりも小さい。

$$n \cdot \sum_{i=1}^{m} 2^{i-1} \cdot \prod_{k=1}^{i} (h(T_k) + 1) .$$

ここで、$h(T_k) + 1$は、最大でもT_kの頂点の数である。 $\prod_i |T_i| =$ |G| である。であるから、これはせいぜい

$$n \cdot \sum_{i=1}^{m} 2^{i-1} \cdot |G| < n \cdot 2^m |G|.$$

ベースmの長さは最大でも$\log |G|$である。

$$O(n \cdot |G|^2).$$

合計所要時間は次のようになる。

$$O(n \cdot |G|^2 + n(^n) \log |G|)。$$

□

12 結びの言葉

本稿では、任意の並べ替えパズルの一般的な解を記述するアルゴリズ

ムを記述することを目的とする。第3節では、このようなアルゴリズムの出力の長さは、並べ替えられる要素の数に対して多項式にはなり得ないことを見た。セクション4では、出力の長さは群順列に線形であることを見た。

定理11.1で見たように、我々が提案したアルゴリズムは、ベースの要素のサイズが一定で、入力の長さを考慮した場合、セットアップ（拡張Schreier-Simsアルゴリズム）と、このセットアップを使って順列パズルを解くために必要な時間とメモリが多項式になる。

(拡張メンバシップ検定アルゴリズム)は、定理11.2で見たように、群次数に対して多項式時間が必要である。定理10.2では、出力の長さが群次数に対して2次関数であることを証明した。

このため、多項式のセットアップを必要とし、直線的な出力長を持つアルゴリズムを見つけるという課題が残る。

参考文献

[1] アポストール、トム・M. 解析的整数論入門.ニューヨーク：Springer-Verlag, 1976.

[2] Babai, László「第 27 章： Automorphism groups, isomorphism, reconstruction".In Graham, R. L.; Gr¨otschel, M.; Lovász, L..組合せ論ハンドブック.アムステルダム：Elsevier, 1995. pp, 1447-1540.

[3] Seress, A´kos.Permutation Group Algorithms. 第 1 版 .Cambridge： Cambridge University Press, 2003.

[4] Turner, Edward C.; Gold, Karen F. "Rubik's Groups" in The American Mathematical Monthly, Vol 92, No 9 (Nov, 1985), pp.